QUASI-OPTIMAL MESH SEQUENCE CONSTRUCTION THROUGH SMOOTHED ADAPTIVE FINITE ELEMENT METHODS

ORNELA MULITA*, STEFANO GIANI†, AND LUCA HELTAI ‡

Abstract. We propose a new algorithm for Adaptive Finite Element Methods (AFEMs) based on smoothing iterations (S-AFEM), for linear, second-order, elliptic partial differential equations (PDEs). The algorithm is inspired by the ascending phase of the V-cycle multigrid method: we replace accurate algebraic solutions in intermediate cycles of the classical AFEM with the application of a prolongation step, followed by a fixed number of *few* smoothing steps. Even though these intermediate solutions are far from the exact algebraic solutions, their a-posteriori error estimation produces a refinement pattern that is substantially equivalent to the one that would be generated by classical AFEM, at a considerable fraction of the computational cost.

We quantify rigorously how the error propagates throughout the algorithm, and we provide a connection with classical a posteriori error analysis. A series of numerical experiments highlights the efficiency and the computational speedup of S-AFEM.

Key word. adaptive mesh refinement, finite element method, second-order elliptic PDEs, a posteriori error analysis, inexact algebraic solution, iterative solvers, smoothing iterations, grid construction.

AMS subject classifications. 65N15, 65N22, 65N30, 65N50, 65N55

1. Introduction.

The efficient numerical simulation of complex real-world phenomena requires the use of computationally affordable discrete models. The adaptive finite element method (AFEM) is one such a scheme for the numerical resolution of partial differential equations (PDEs) in computational sciences and engineering. In finite element simulations (FEM), the domain of a PDE is discretized into a large set of small and simple domains (the cells or elements) depending on a size parameter $h > 0$. Typical shapes that are used for the discretization are triangles, quadrilaterals, tetrahedrons, or hexahedrons. The solution space is constructed by gluing together simpler finite dimensional spaces, defined on a piecewise manner on each cell, and the original problem is solved on this simpler, finite dimensional space, transforming the original PDE into an algebraic system of equations. Rigorous analysis of the numerical method allows one to estimate the discretization error both a priori (giving global bounds on the total error that depend on a global size parameter h), and a posteriori (providing a local distribution of the error on the discretized mesh in terms of known quantities). Classical AFEM consists of successive loops of the steps

$$\mathcal{S}olve \longrightarrow \mathcal{E}stimate \longrightarrow \mathcal{M}ark \longrightarrow \mathcal{R}efine \tag{1.1}$$

to decrease the total discretization error, by repeating the FEM solution process ($\mathcal{S}olve$) on a mesh that has been refined ($\mathcal{R}efine$) on areas where the a-posteriori analysis ($\mathcal{E}stimate$) has shown that the error is larger ($\mathcal{M}ark$).

Intermediate solution steps are *instrumental* for the construction of the finally adapted grid, and play no role in the final solution, which is the only one which is retained for analysis and processing.

In this work we present and analyze a simple yet effective algorithm to reduce the overall computational cost of the AFEM algorithm, by providing a fast procedure

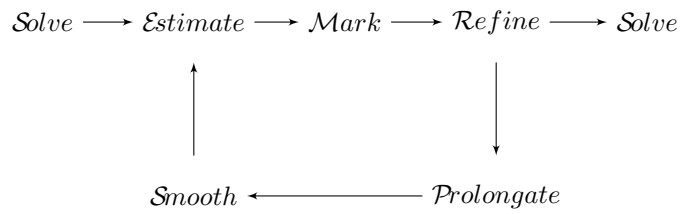
*Corresponding Author: omulita@sissa.it, Scuola Internazionale Superiore di Studi Avanzati, Via Bonomea 265, 34136 Trieste, IT

†stefano.giani@durham.ac.uk, Durham University, Stockton Road, Durham, DH1 3LE UK

‡luca.heltai@sissa.it, Scuola Internazionale Superiore di Studi Avanzati, Via Bonomea 265, 34136 Trieste, IT

for the construction of a quasi-optimal mesh sequence that does not require the exact solution of the algebraic problem in the intermediate steps of AFEM.

We propose a novel *Smoothed Adaptive Finite Element* algorithm (S-AFEM), where the accurate $\mathcal{S}olve$ step for all intermediate loops is replaced by the application of a prolongation step ($\mathcal{P}rolongate$), followed by a fixed number of *few* smoothing steps ($\mathcal{S}mooth$):



In our setting, intermediate steps have non-negligible algebraic errors due to the inexact solution of the linear system. This is in contrast with the common practical assumption made in AFEM, where it is assumed that the $\mathcal{S}olve$ step produces the *exact* solution of the algebraic system. Recent developments dedicated a great deal of effort to account for inexactness of the algebraic approximations and introduce stopping criteria based on the interplay between discretization and algebraic computation in adaptive FEM. Among others, we mention the seminal contributions [10, 26, 4, 5, 34, 30, 33, 32, 21, 31, 29, 20].

Nevertheless, most of this literature focuses on ways to estimate the algebraic error, without really exploiting the other side of the coin: inexact approximate solutions, with large algebraic error, may still offer large computational savings when used in the correct way.

The algebraic errors of the intermediate steps are the price we pay to improve the speed of the overall algorithm. We analyze how these errors propagate through the $\mathcal{E}stimate$-$\mathcal{M}ark$-$\mathcal{R}efine$ steps, and we show that the final grid sequence obtained through S-AFEM provides the same accuracy of the one obtained through classical AFEM, at a fraction of the computational cost.

S-AFEM takes its inspiration from the ascending phase of the V-cycle multigrid method, where a sequence of prolongation and smoothing steps is applied to what is considered an algebraically exact solution at the coarsest level. In the multigrid literature, this procedure is used to transfer the low frequency information contained in the coarse solution to a finer –nested– grid, where some steps of a smoothing iteration are applied in order to improve the accuracy of the solution in the high frequency range. We refer to the classical books [24, 25, 39, 13, 12] for a more in-depth analysis of multigrid methods. The iteration of this procedure turns out to be very effective in providing accurate algebraic solutions in $O(N)$ time, where N is the dimension of the final algebraic system. This procedure is based on the principle that even a small number of smoothing iterations is sufficient to eliminate the high frequency error, while the prolongation from coarser grids guarantees the convergence in the low frequency regime, resulting in an overall accurate solution.

However, there is a main difference between the ascending phase of the V-cycle multigrid method and AFEM: in AFEM the next grid in the sequence is unknown, and requires an exact algebraic solution on the current grid to trigger the $\mathcal{E}stimate$-$\mathcal{M}ark$-$\mathcal{R}efine$ steps.

The motivation behind the strategy at the base of S-AFEM is that classical

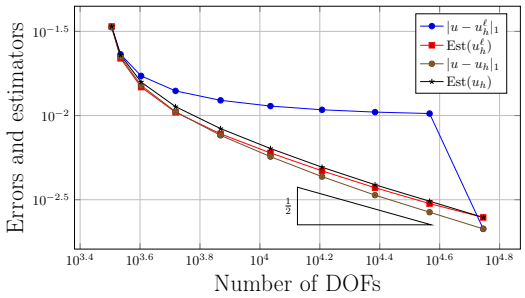


FIG. 1.1. *The values of the total error energy norm and of the error estimator for each loop of the classical AFEM ($|u-u_h|_1$ and $Est(u_h)$) and S-AFEM with $\ell = 2$ smoothing iterations ($|u-u_h^\ell|_1$ and $Est(u_h^\ell)$) for the classical L-shaped domain problem in 2D. The first and last loop are solved exactly by both methods.*

residual-based a posteriori error estimators [38] used in the $\mathcal{E}stimate$ step are not sensitive to low frequencies in the solution, as shown in Figure 1.1 for a benchmark example. Consequently, their application to very inaccurate approximate solutions in intermediate loops – only capturing high frequency oscillations through a smoother – produces an equally good grid refinement pattern at each loop, at a fraction of the computational cost.

The first and last loops of the S-AFEM algorithm coincide with those of the classical AFEM. In intermediate S-AFEM loops, however, the solution is far from the exact algebraic solution. These intermediate solutions serve solely to the construction of the final grid, and find no other use in the final computations, therefore their inexactness is irrelevant, provided that the finally adapted grid provides a good approximation. Their role is *instrumental* in triggering the $\mathcal{E}stimate - \mathcal{M}ark - \mathcal{R}efine$ steps.

We show that the a-posteriori error estimator applied to the intermediate approximations triggers a $\mathcal{M}ark$ step where the refinement pattern is substantially equivalent to the one that would be generated by a classical $\mathcal{S}olve$ step, at a considerable fraction of the computational cost. For three dimensional problems, the speedup in the intermediate loops is in the hundreds, and even if the final grid is not exactly identical to the one that would be obtained with the classical AFEM, the accuracy of the final solutions is comparable.

The article is organised as follows. We start by introducing a general multilevel framework applied to a simple model problem in Section 2. Its algebraic properties are analyzed in detail in Section 3. Section 4 is dedicated to the connection between the algebraic error and the a posteriori error estimates. Section 5 provides a detailed description of the S-AFEM algorithm, while Sections 6 and 7 present some numerical examples that show the efficiency and the computational speedup of S-AFEM, and provide some conclusions.

2. Multilevel framework. In this section we describe the Poisson model problem and discuss its algebraic resolution in a multilevel framework. In particular, we motivate the reason behind the use of smoothing iterations applied to the prolongation of the approximation from previous levels. This simple, but significant model problem serves as a prototype for the larger class of linear elliptic, second-order, boundary value problems (BVPs).

The variational formulation of such problems reads: seek $u \in V$ s.t. $\mathcal{A}u = f$ in V under suitable boundary conditions, where $(V, \|\bullet\|)$ is a normed Hilbert space defined over a Lipschitz bounded domain Ω, the linear operator $\mathcal{A} : V \to V^\star$ is a second order elliptic operator, and $f \in V^\star$ is a given datum. FEM provides numerical solutions to the above problem in a finite dimensional solution space $V_h \subset V$, typically made up by continuous and piecewise polynomial functions, and transforms the continuous

problem above in a discrete model of type $\mathcal{A}_h u_h = f_h$ in V_h under suitable boundary conditions, where e.g., $\mathcal{A}_h = \mathcal{A}\,|_{V_h}$. The overall procedure leads to the resolution of a (potentially very large) linear algebraic system of equations of type $A\mathbf{u} = \mathbf{f}$ in $\mathbb{R}^N$, where $N = \dim(V_h)$.

2.1. Model Problem. As a model problem, we consider Poisson's equation with homogeneous Dirichlet boundary conditions. Let $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ be a bounded, polygonal domain (an open and connected set with polygonal boundary) with Lebesgue and Sobolev spaces $L^2(\Omega)$ and $H_0^1(\Omega)$. We look for the solution $u \in H_0^1(\Omega)$ such that

$$-\Delta u = f \text{ in } \Omega \text{ and } u = 0 \text{ on } \Gamma := \partial\Omega, \tag{2.1}$$

where $f \in L^2(\Omega)$ is a given source term. We use the standard notation for norms and scalar products in Lebesgue and Sobolev spaces (cf. [1]): for $u \in H_0^1(\Omega)$ and $\omega \subset \Omega$, we write $|u|_{1,\omega} := (\int_\omega |\nabla u|^2)^{1/2}$ and denote by $(\cdot,\cdot)_\omega$ the $L^2(\omega)$- scalar product with corresponding norm $\|\cdot\|_\omega$. For $\omega = \Omega$, we omit the corresponding subscripts. The weak form of (2.1) is to find $u \in H_0^1(\Omega)$ s.t.

$$(\nabla u, \nabla v) = (f, v)\ \forall v \in H_0^1(\Omega). \tag{2.2}$$

We consider a shape regular family of triangulations $\{\mathcal{T}_h\}_h$ of Ω in the sense of Ciarlet [18], depending on a parameter $h > 0$ with shape regularity parameter $C_{\mathcal{T}_h}$. We will consider triangulations consisting of triangles or convex quadrilaterals in two dimensions, and tetrahedrons or convex hexahedrons in three dimensions; we denote them by T and we generically call them *cells*.

We denote by z the nodes of $\mathcal{T}_h$ (i.e. the vertices of the cells) and by $\mathcal{N}_h$ the set of all nodes, while $\mathcal{N}_{h,int}$ denotes the set of the free nodes. The set of all edges/faces E of the cells is denoted by $\mathcal{E}_h$ and similarly, $\mathcal{E}_{h,int} := \mathcal{E}_h \setminus \Gamma$ is the set of internal edges/faces. Let φ_z be the nodal basis function associated to a node $z \in \mathcal{N}_h$ with support ω_z, which is equal to the patch $\omega_z = \cup\{T \in \mathcal{T}_h | z \in T\}$. We use the Courant finite element space $V_h := \text{span}\{\varphi_z | z \in \mathcal{N}_{h,int}\} \subset H_0^1(\Omega)$. The discrete approximation $u_h \in V_h$ is called a *Galerkin solution* and it is defined by the discrete system

$$(\nabla u_h, \nabla v_h) = (f, v_h)\ \forall v_h \in V_h. \tag{2.3}$$

In exact arithmetic, the discretization error $e_h := u - u_h$ satisfies the standard orthogonality condition

$$(\nabla(u - u_h), \nabla v_h) = 0\ \forall v_h \in V_h. \tag{2.4}$$

Equation (2.4) is the basic relation under which classical a posteriori error bounds for the discretization error are derived (cf. Section 4).

Let $N = \dim(V_h)$, the discrete system (2.3) leads to a linear algebraic system of type

$$A\mathbf{u} = \mathbf{f} \text{ in } \mathbb{R}^N, \tag{2.5}$$

where A denotes the symmetric positive definite (SPD) stiffness matrix with entries $a_{ij} := (\nabla\varphi_j, \nabla\varphi_i)\ \forall\, i, j = 1, .., N$, $\mathbf{u} = [u_1, \ldots, u_N]^T$ denotes the coefficients vector in

$\mathbb{R}^N$ of the discrete approximation $u_h = \sum_{j=1}^N u_j \varphi_j \in V_h$ and $\mathbf{f} = [f_1, \dots, f_N]^T$ is the vector with entries $f_j = (f, \varphi_j)\ \forall j = 1, .., N$.

In our framework, we will consider a nested sequence of shape regular triangulations $\mathcal{T}_k$, for $k = 1, \dots, \bar{k}$, which induces a nested sequence of finite element spaces

$$V_1 \subset V_2 \subset \cdots \subset V_{\bar{k}} \subset H_0^1(\Omega). \tag{2.6}$$

Typical examples are the ones generated during global and local mesh-refinement techniques, starting from a given (coarse) uniform triangulation $\mathcal{T}_1$.

REMARK 2.0.1. *Relation (2.6) does not hold true for all adaptive refinements. In particular, refinement procedures involving red-green refinements do not satisfy this. In this work we use a code based on the open source library* `deal.II`*, that handles local refinement through hanging nodes (see [9, 7, 3, 6]), and this condition is always satisfied if no de-refinement is applied. This will be the case for the numerical tests described in Section 6.*

We let $N_k := \dim(V_k)$, for $k = 1, \dots, \bar{k}$. By construction, the inequalities $N_1 < N_2 < \cdots < N_{\bar{k}}$ hold true. The associated discrete systems for each level $k = 1, 2, \dots, \bar{k}$ read

$$(\nabla u_k, \nabla v_k) = (f, v_k)\ \ \forall\, v_k \in V_k \tag{2.7}$$

and they generate linear systems of type

$$A_k \mathbf{u}_k = \mathbf{f}_k \tag{2.8}$$

of respective dimensions N_k.

2.2. Smoothed multilevel methods. For the algebraic resolution of systems (2.8), we will consider smoothing iterations. We recall that by the Spectral Theorem, the eigenvectors of the stiffness matrices A_k form an orthonormal basis of $\mathbb{R}^{N_k}$ [27], therefore the error after any iteration that solves (2.8) can be decomposed in this basis. The corresponding eigenvalues are ordered non-decreasingly. It is well known that eigenvectors corresponding to larger eigevenvalues are increasingly oscillatory. Smoothing iterations have the characteristic to damp the components of the error in the directions of the most oscillatory eigenvectors.

For simplicity of exposition, in this work we only use Richardson iteration as a smoothing iteration, but other choices are possible, see, for example, the review in [12, 23, 39].

Given $\omega_k \in \mathbb{R}$ a fixed parameter and $\mathbf{u}_k^{(0)} \in \mathbb{R}^{N_k}$ an initial guess, Richardson iteration for the resolution of (2.8) takes the form

$$\mathbf{u}_k^{(i+1)} = \mathbf{u}_k^{(i)} + \omega_k(\mathbf{f}_k - A_k \mathbf{u}_k^{(i)})\ \text{ for } i = 0, 1, \dots \tag{2.9}$$

From equation (2.9) it is immediate that the error after $i+1$ iterations $\mathbf{e}_k^{(i+1)}$ satisfies the error propagation formula $\mathbf{e}_k^{(i+1)} = M_k \mathbf{e}_k^{(i)} = \cdots = M_k^{i+1} \mathbf{e}_k^{(0)}$, where $M_k := Id_{N_k} - \omega_k A_k$ is the iteration matrix of the method. The optimal choice of the parameter ω_k is $\omega_k = 1/\gamma_k$, where γ_k is a *damping parameter* of the same order of the spectral radius of A_k. We refeer to the classical books [24, 35] for more details on the Richardson iteration.

Due to the error propagation formula, it can be easily seen that the error components after a single Richardson iteration are reduced by a factor which is "close to zero" in correspondence to the more oscillatory eigenvectors and "close to one" in correspondence to the less oscillatory eigenvectors. As a consequence, after a single Richardson iteration, the high frequency components of the error, corresponding to the more oscillatory eigenvectors, will have been strongly reduced.

This characteristic of Richardson iteration makes it a good smoother candidate for many multilevel algorithms, where one solves exactly on a coarse grid (reaching convergence in all components), and then performs a sequence of prolongations followed by a fixed number of few smoothing steps, to improve convergence in the finer grids, under the assumption that lower frequencies have already been taken care of in the previous levels.

The prolongation is achieved by considering the canonical embedding $i_k^{k+1} : V_k \hookrightarrow V_{k+1}$ that embeds functions $u_k \in V_k$ in the space V_{k+1}. We denote by $I_k^{k+1} : \mathbb{R}^{N_k} \to \mathbb{R}^{N_{k+1}}$ the corresponding discrete matrix. Notice that the matrix representation won't be the identity matrix, since we're using different basis functions in V_k and in V_{k+1}. As an example, consider linear finite element functions. These are uniquely determined by their values in the nodes. For nodes that exist both in $\mathcal{T}_k$ and $\mathcal{T}_{k+1}$, the value at those nodes can be determined in $\mathcal{T}_k$ and it remains the same. For the nodes in $\mathcal{T}_{k+1}$ that are not in $\mathcal{T}_k$, their values are determined by linear interpolation. We will refer to multilevel algorithms that adopt the above procedure of resolution as smoothed-multilevel methods.

The effect of the prolongation is that it allows the iterative solver to start from an already good approximation of the solution in its low frequency part. In this work, we will make the (inexact) assumption that the prolongation operation leaves unaltered the low frequency components of vectors defined from the previous mesh (cf. Subsection 3.2). Then, by applying smoothing iterations, we're converging towards the solution in the highest frequencies. Despite being far less competitive as solvers for large systems in general, smoothing iterations turn out to be very useful in this context, similarly to what happens in multigrid methods.

3. Algebraic Error Analysis.

In this section we analyze the algebraic error propagation in smoothed-multilevel methods. More precisely, we assume that the algebraic system at the first level is solved arbitrarily, then we apply prolongations between successive levels followed by a fixed number of smoothing steps. We first provide a one step recursive formula for the error propagation. Afterwards, we introduce the frequency-coupling and smoothing (FCS) matrices and derive a compact formula for the propagation of the error across smoothing steps and mesh prolongations. Finally, we provide the algebraic error analysis under the assumption that the prolongation operator preserves low frequencies from the previous level.

3.1. Error propagation.

THEOREM 3.1 (Error propagation). *Let $\mathbf{e}_k^{(\ell)}$ and $\mathbf{e}_{k+1}^{(\ell)}$ denote the algebraic errors after ℓ smoothing iterations respectively at step k and $k+1$, for $k = 1, \dots, \bar{k} - 1$. Let*

$$\mathbf{a}_{k+1} := \mathbf{u}_{k+1} - I_k^{k+1}\mathbf{u}_k \in \mathbb{R}^{N_{k+1}} \tag{3.1}$$

denote the difference between the exact algebraic solution $\mathbf{u}_{k+1}$ at level $k+1$ and the prolongation of the exact algebraic solution $\mathbf{u}_k$ from the previous level k to the current level $k+1$, for $k = 1, \dots, \bar{k}-1$. Notice that the vector $\mathbf{a}_1$ is not defined, so that definition (3.1) starts from the vector $\mathbf{a}_2$. Then, the following error propagation recursive

formula holds true

$$\mathbf{e}_{k+1}^{(\ell)} = {M_{k+1}}^{\ell}(\mathbf{a}_{k+1} + I_k^{k+1}\mathbf{e}_k^{(\ell)}), \quad \textit{for} \;\; k = 1, \dots, \bar{k} - 1. \tag{3.2}$$

Proof. Let $\mathbf{e}_1 = \mathbf{u}_1 - \mathbf{u}_1^c$ be the error after the first cycle $k = 1$, where $\mathbf{u}_1^c$ is the numerical computed approximation. After prolongating $\mathbf{u}_1^c$ to the next level $k = 2$, there is an initial error

$$\begin{aligned} \mathbf{e}_2^{(0)} &= \mathbf{u}_2 - I_1^2\mathbf{u}_1^c \\ &= \mathbf{u}_2 - I_1^2\mathbf{u}_1 + I_1^2\mathbf{e}_1 \\ &= \mathbf{a}_2 + I_1^2\mathbf{e}_1. \end{aligned} \tag{3.3}$$

After ℓ smoothing iterations there is produced a smoothed approximation $\mathbf{u}_2^{(\ell)}$ and the final error is given by

$$\begin{aligned} \mathbf{e}_2^{(\ell)} &= {M_2}^{\ell}\mathbf{e}_2^{(0)} \\ &= {M_2}^{\ell}\mathbf{a}_2 + {M_2}^{\ell}I_1^2\mathbf{e}_1. \end{aligned} \tag{3.4}$$

Let now $k = 2, 3, \dots, \bar{k} - 1$ be generic. We prolongate the smoothed approximation $\mathbf{u}_k^{(\ell)} = \mathbf{u}_k - \mathbf{e}_k^{(\ell)}$ from step k to obtain the initial guess for step $k + 1$

$$\begin{aligned} \mathbf{u}_{k+1}^{(0)} &= I_k^{k+1}\mathbf{u}_k^{(\ell)} \\ &= I_k^{k+1}\mathbf{u}_k - I_k^{k+1}\mathbf{e}_k^{(\ell)}, \end{aligned} \tag{3.5}$$

which produces the initial error

$$\begin{aligned} \mathbf{e}_{k+1}^{(0)} &= \mathbf{u}_{k+1} - \mathbf{u}_{k+1}^{(0)} \\ &= \mathbf{u}_{k+1} - I_k^{k+1}\mathbf{u}_k + I_k^{k+1}\mathbf{e}_k^{(\ell)} \\ &= \mathbf{a}_{k+1} + I_k^{k+1}\mathbf{e}_k^{(\ell)}. \end{aligned} \tag{3.6}$$

After ℓ smoothing iterations the final error at step $k + 1$ is

$$\begin{aligned} \mathbf{e}_{k+1}^{(\ell)} &= {M_{k+1}}^{\ell}\mathbf{e}_{k+1}^{(0)} \\ &= {M_{k+1}}^{\ell}(\mathbf{a}_{k+1} + I_k^{k+1}\mathbf{e}_k^{(\ell)}), \end{aligned} \tag{3.7}$$

which proves the recursive formula. ∎

Observation 3.2. *If we repetitively apply the one-step error propagation equation* (3.7)*, we get a recursion of the type*

$$\begin{aligned} \mathbf{e}_{k+1}^{(\ell)} &= {M_{k+1}}^{\ell}(\mathbf{a}_{k+1} + I_k^{k+1}\mathbf{e}_k^{(\ell)}) \\ &= {M_{k+1}}^{\ell}(\mathbf{a}_{k+1} + I_k^{k+1}({M_k}^{\ell}(\mathbf{a}_k + I_{k-1}^{k}\mathbf{e}_{k-1}^{(\ell)}))) \\ &\dots \\ &= {M_{k+1}}^{\ell}(\mathbf{a}_{k+1} + I_k^{k+1}{M_k}^{\ell}(\mathbf{a}_k + I_{k-1}^{k}{M_{k-1}}^{\ell}(\mathbf{a}_{k-1} + \dots))). \end{aligned} \tag{3.8}$$

By applying all the multiplications extensively we get the following extended error propagation formula for smoothed-multilevel methods

$$\begin{aligned} \mathbf{e}_{k+1}^{(\ell)} = {M_{k+1}}^{\ell}\mathbf{a}_{k+1} + {M_{k+1}}^{\ell}I_k^{k+1}{M_k}^{\ell}\mathbf{a}_k + {M_{k+1}}^{\ell}I_k^{k+1}{M_k}^{\ell}I_{k-1}^{k}{M_{k-1}}^{\ell}\mathbf{a}_{k-1} + \dots \\ \cdots + {M_{k+1}}^{\ell}I_k^{k+1}{M_k}^{\ell}I_{k-1}^{k}{M_{k-1}}^{\ell}\cdots{M_3}^{\ell}I_2^3{M_2}^{\ell}\mathbf{a}_2 \\ + {M_{k+1}}^{\ell}I_k^{k+1}{M_k}^{\ell}I_{k-1}^{k}{M_{k-1}}^{\ell}\cdots{M_3}^{\ell}I_2^3{M_2}^{\ell}I_1^2\mathbf{e}_1. \end{aligned} \tag{3.9}$$

OBSERVATION 3.3. *If we let*

$$\begin{aligned}\mathbf{a} := \mathbf{a}_{k+1} + I_k^{k+1} {M_k}^{\ell} \mathbf{a}_k + I_k^{k+1} {M_k}^{\ell} \dots I_{k-1}^{k} {M_{k-1}}^{\ell} \mathbf{a}_{k-1} \\ \dots + I_k^{k+1} {M_k}^{\ell} I_{k-1}^{k} {M_{k-1}}^{\ell} \cdots {M_3}^{\ell} I_2^3 {M_2}^{\ell} I_1^2 \mathbf{e}_1,\end{aligned} \tag{3.10}$$

then equation (3.9) *becomes*

$$\mathbf{e}_{k+1}^{(\ell)} = {M_{k+1}}^{\ell} \mathbf{a}, \tag{3.11}$$

which means that the algebraic error at any step $k+1$ is the result of ℓ smoothing iterations applied to the vector $\mathbf{a}$ that defines the error accumulated from prolongating the contribution of the algebraic errors coming from all previous steps.

DEFINITION 3.4 (Frequency-coupling and smoothing (FCS) matrices). *Define the frequency-coupling and smoothing (FCS) matrix*

$$B_{j+1,j} := I_j^{j+1} M_j^{\ell} \in \mathbb{R}^{N_{j+1} \times N_j} \text{ for } j = 2, \dots, k. \tag{3.12}$$

and the frequency-coupling and smoothing product (FCSP)

$$\mathbb{B}_{k+1,i} := B_{k+1,k} \dots B_{i+1,i} \in \mathbb{R}^{N_{k+1} \times N_i} \text{ for } i = 2, \dots, k. \tag{3.13}$$

THEOREM 3.5 (Error propagation formula for smoothed-multilevel methods). *The algebraic error in smoothed-multilevel methods satisfies the following error propagation formula for any step k, for $k = 2, \dots, \bar{k} - 1$*

$$\mathbf{e}_{k+1}^{(\ell)} = {M_{k+1}}^{\ell} \left(\mathbf{a}_{k+1} + \sum_{j=2}^{k} \mathbb{B}_{k+1,j} \mathbf{a}_j + \mathbb{B}_{k+1,2} I_1^2 \mathbf{e}_1 \right), \tag{3.14}$$

where the vectors $\mathbf{a}_j$ are defined by (3.1).

Proof. The proof is a trivial consequence of substituting Definition 3.4 in the extended error propagation formula (3.9), which gives

$$\begin{aligned}\mathbf{e}_{k+1}^{(\ell)} &= {M_{k+1}}^{\ell} (\mathbf{a}_{k+1} + B_{k+1,k} \mathbf{a}_k + \dots \\ &\quad \dots + B_{k+1,k} B_{k,k-1} \dots B_{3,2} \mathbf{a}_2 + B_{k+1,k} B_{k,k-1} \\ &\quad \dots B_{3,2} I_1^2 \mathbf{e}_1) \\ &= {M_{k+1}}^{\ell} (\mathbf{a}_{k+1} + \mathbb{B}_{k+1,k} \mathbf{a}_k + \mathbb{B}_{k+1,k-1} \mathbf{a}_{k-1} + \\ &\quad \dots + \mathbb{B}_{k+1,2} \mathbf{a}_2 + \mathbb{B}_{k+1,2} I_1^2 \mathbf{e}_1).\end{aligned} \tag{3.15}$$

∎

Let $\{\mathbf{w}_j^{(i)}\}_{i=1}^{N_j}$ be the eigenvectors of the stiffness matrix A_j at level j. From the discussion in Subsection 2.2, any vector $\mathbf{v} \in \mathbb{R}^{N_j}$ can be uniquely decomposed as $\mathbf{v} := \sum_{i=1}^{N_j} v_i \mathbf{w}_j^{(i)}$. The low frequency components of $\mathbf{v}$ are those from 1 to $N_j/2$, corresponding to the less oscillatory eigenvectors, while the high frequency components of $\mathbf{v}$ are the ones from $N_j/2 + 1$ to N_j, corresponding to the more oscillatory eigenvectors [35, 24].

Next, we define the frequency cutoff projection operators, which are a useful tool to analyze the structure of the FCS matrix $B_{j+1,j}$. In Theorem 3.7 we provide a decomposition of the FCS matrix as the product of the prolongation matrix I_j^{j+1} with the low frequency cutoff projection operator and another matrix, which has a contraction effect on the norms of the vectors.

DEFINITION 3.6 (Frequency cutoff operators). *We define the projection operator* $L_j : \mathbb{R}^{N_j} \to \mathbb{R}^{N_j}$, *such that* $\mathbf{v} \mapsto L_j\mathbf{v} := \sum_{i=1}^{N_j/2} v_i \mathbf{w}_i^{(j)}$ *and the projection operator* $H_j : \mathbb{R}^{N_j} \to \mathbb{R}^{N_j}$, *such that* $\mathbf{v} \mapsto H_j\mathbf{v} := \sum_{i=N_j/2+1}^{N_j} v_i \mathbf{w}_i^{(j)}$.

In particular,

$$L_j + H_j = Id_{N_j} \text{ and } \|\mathbf{v}\|^2 = \|L_j\mathbf{v}\|^2 + \|H_j\mathbf{v}\|^2 \quad \forall \mathbf{v} \in \mathbb{R}^{N_j}. \tag{3.16}$$

THEOREM 3.7 (Structure of the FCS matrix). *Let* $j = 2, \ldots, k$ *be fixed. The FCS matrix can be decomposed as*

$$B_{j+1,j} = I_j^{j+1} L_j + C_j, \tag{3.17}$$

where the matrix $C_j \in \mathbb{R}^{N_j+1 \times N_j}$ *is defined as*

$$C_j := I_j^{j+1}((M_j{}^{\ell} - Id_{N_j})L_j + M_j{}^{\ell} H_j) \tag{3.18}$$

and satisfies

$$\|C_j\mathbf{v}\| \leq c\|\mathbf{v}\| \quad \forall \mathbf{v} \in \mathbb{R}^{N_j}, \text{ where } c < 1. \tag{3.19}$$

Proof. We apply definition (3.12) of the FCS matrix and relation (3.16) and we get

$$\begin{aligned} B_{j+1,j} &= I_j^{j+1} M_j{}^{\ell} \\ &= I_j^{j+1} M_j{}^{\ell} L_j + I_j^{j+1} M_j{}^{\ell} H_j \\ &= I_j^{j+1} L_j + (I_j^{j+1} M_j{}^{\ell} L_j - I_j^{j+1} L_j + I_j^{j+1} M_j{}^{\ell} H_j) \\ &= I_j^{j+1} L_j + C_j, \end{aligned} \tag{3.20}$$

where $C_j := I_j^{j+1}((M_j{}^{\ell} - Id_{N_j})L_j + M_j{}^{\ell} H_j)$.

Next, in order to prove (3.19), consider $\mathbf{v} \in \mathbb{R}^{N_j}$ and estimate

$$\begin{aligned} \|C_j\mathbf{v}\|^2 &= \|I_j^{j+1}((M_j{}^{\ell} - Id_{N_j})L_j + M_j{}^{\ell} H_j)\mathbf{v}\|^2 \\ &= \|((M_j{}^{\ell} - Id_{N_j})L_j + M_j{}^{\ell} H_j)\mathbf{v}\|^2 \\ &\leq (\|(M_j{}^{\ell} - Id_{N_j})L_j\mathbf{v}\| + \|M_j{}^{\ell} H_j\mathbf{v}\|)^2 \\ &\leq 2(\|(M_j{}^{\ell} - Id_{N_j})L_j\mathbf{v}\|^2 + \|M_j{}^{\ell} H_j\mathbf{v}\|^2), \end{aligned} \tag{3.21}$$

where we've applied the triangle inequality and the discrete Cauchy-Schwarz inequality.

In order to analyze the terms in the rhs of (3.21), we decompose $\mathbf{v}$ in the orthonormal basis of eigenvectors $\{\mathbf{w}_j^{(i)}\}_{i=1}^{N_j}$ of A_j. By construction, these are also the eigenvectors of the Richardson iteration matrix $M_j = Id_{N_j} - \omega_j A_j$. Therefore, any time we apply M_j to $\mathbf{v}$, we get $M_j\mathbf{v} = M_j\left(\sum_{i=1}^{N_j} v_i \mathbf{w}_j^{(i)}\right) = \sum_{i=1}^{N_j} v_i (M_j \mathbf{w}_j^{(i)}) = \sum_{i=1}^{N_j} \theta_j^{(i)} v_i \mathbf{w}_j^{(i)}$, where $\theta_j^{(i)}$ is the eigenvalue of M_j corresponding to $\mathbf{w}_j^{(i)}$. It is well known [35, 25] that $\theta_j^{(i)}$ is close to 1 for $i \in [1, N_j/2]$, i.e., for low frequency components and close to 0 for $i \in (N_j/2, N_j]$, i.e., for high frequency components.

For the first term in the rhs of (3.21) we estimate

$$
\begin{aligned}
\|({M_j}^{\ell} - Id_{N_j})L_j\mathbf{v}\|^2 &= \left\| \sum_{i=1}^{N_j/2} \left((\theta_j^{(i)})^{\ell} - 1 \right) v_i \mathbf{w}_j^{(i)} \right\|^2 \\
&= \sum_{i=1}^{N_j/2} \left((\theta_j^{(i)})^{\ell} - 1 \right)^2 v_i^2 \\
&\leq \left((\theta_j^{(N_j/2)})^{\ell} - 1 \right)^2 \sum_{i=1}^{N_j/2} v_i^2 \\
&= \left((\theta_j^{(N_j/2)})^{\ell} - 1 \right)^2 \|L_j\mathbf{v}\|^2.
\end{aligned} \tag{3.22}
$$

Likewise,

$$
\begin{aligned}
\|{M_j}^{\ell} H_j\mathbf{v}\|^2 &= \left\| \sum_{i=N_j/2+1}^{N_j} (\theta_j^{(i)})^{\ell} v_i \mathbf{w}_j^{(i)} \right\|^2 \\
&= \sum_{i=N_j/2+1}^{N_j} (\theta_j^{(i)})^{2\ell} v_i^2 \\
&\leq (\theta_j^{(N_j/2+1)})^{2\ell} \sum_{i=N_j/2+1}^{N_j} v_i^2 \\
&= (\theta_j^{(N_j/2+1)})^{2\ell} \|H_j\mathbf{v}\|^2.
\end{aligned} \tag{3.23}
$$

We let $c := 2\max\left\{ \left((\theta_j^{(N_j/2)})^{\ell} - 1 \right)^2, (\theta_j^{(N_j/2+1)})^{2\ell} \right\} < 1$, we substitute it into (3.21), and we get estimate (3.19). ∎

3.2. A qualitative analysis: non-interacting frequency coupling Hypothesis. It is safe to state that local refinement in finite element simulations introduces more frequencies in the higher part of the spectrum, perturbing only slightly the lowest part of the spectrum. In order to give a qualitative interpretation to the error propagation formula, we reinterpret Theorem 3.5 under the following (inexact) assumption.

ASSUMPTION 3.8 (Non-interacting frequency coupling Hypothesis for smoothed-multilevel methods). *We assume that*

$$
L_j I_{j-1}^j L_{j-1} = I_{j-1}^j L_{j-1} \ \forall j = 1, \dots k, \tag{3.24}
$$

i.e. the prolongation operator preserves low frequencies from the previous level.

It is clear that this can only hold up to approximation order (in an a priori setting) or up to the a posteriori error estimate for the low frequency eigenfunctions in an adaptive setting. However, it is still useful to derive the propagation formula for the algebraic error under Assumption 3.8 to understand how the algebraic error propagates throughout the algorithm. For the proof, we take advantage of the decomposition of the FCS matrix given by Theorem 3.7 to obtain a decomposition for the FCSP (3.13) and we substitute the results in Theorem 3.5.

THEOREM 3.9 (Error propagation formula for smoothed-multilevel methods under the non-interacting frequency coupling Hypothesis). *The algebraic error in smoothed-multilevel methods satisfies the following error propagation formula for any step k, for $k = 2, \dots, \bar{k} - 1$*

$$\mathbf{e}_{k+1}^{(\ell)} = {M_{k+1}}^{\ell} \left(\mathbf{a}_{k+1} + \sum_{j=2}^{k} I_j^{k+1} L_j \mathbf{a}_j + I_2^{k+1} L_2 I_1^2 \mathbf{e}_1 + \sum_{j=2}^{k} \mathbb{D}_{k+1,j} \mathbf{a}_j + \mathbb{D}_{k+1,2} I_1^2 \mathbf{e}_1 \right), \tag{3.25}$$

where the matrix $\mathbb{D}_{k,m} \in \mathbb{R}^{N_k \times N_m}$ is such that $\|\mathbb{D}_{k,m} v\| \leq c\|v\|$, $\forall v \in \mathbb{R}^{N_m}$, where $c < 1$, the matrix $I_m^k := I_{k-1}^k \dots I_m^{m+1}$, $\forall k > m+1$ and the vectors $\mathbf{a}_j$ are defined by (3.1), i.e., $\mathbf{a}_j := \mathbf{u}_j - I_{j-1}^j \mathbf{u}_{j-1}$.

Proof. Observing that

$$\mathbb{B}_{k,m} = B_{k,k-1} \mathbb{B}_{k-1,m}, \qquad \forall k > m+1, \tag{3.26}$$

we can recursively apply the decomposition of $B_{k,k-1}$, and, using hypothesis (3.24), we conclude that

$$\mathbb{B}_{k,m} = I_m^k L_m + \mathbb{D}_{k,m}, \tag{3.27}$$

where the matrix $\mathbb{D}_{k,m}$ is in $\mathbb{R}^{N_k \times N_m}$, and contains all the mixed products. In particular, in every one of these products, there will always be at least one of the matrices C_j for some j, that is,

$$\|\mathbb{D}_{k,m} v\| \leq c\|v\|, \ \forall v \in \mathbb{R}^{N_m}, \tag{3.28}$$

where $c < 1$. We conclude by substituting decomposition (3.27) to the error propagation formula (3.14) in Theorem 3.5. ∎

REMARK 3.9.1. *Theorem 3.9 quantifies the algebraic error that is accumulated after $k+1$ cycles in smoothed-multilevel methods, under the assumption that low frequencies are preserved by the prolongation operator. The smoothing matrix at cycle $k+1$ is responsible for dumping the most oscillatory part of this error. There is a contribution given by the accumulation of all low frequency-parts of the errors of all previous cycles (c.f. second and third term in the summation in the rhs of (3.25)), which is expected to be "small", since low frequencies of the exact algebraic solution at a mesh are close to the low frequencies of the exact algebraic solution at the successive mesh. Finally there is a last type of contribution, which is given by mixed products (cf. fourth and fifth term in the summation in the rhs of (3.25)), which is also "small" due to $c < 1$.*

Assumption 3.8 is useful to derive a formula that identifies qualitatively how the error propagates between successive refinement levels, by distinguishing between high and low frequency parts in the error propagation formula for smoothed-multilevel methods (3.14). However, it is not essential for its proof, given in Theorem 3.5. It can be interpreted as a condition on the distribution of the degrees of freedoms between grids on different levels. In particular, it implies that all low eigenfunctions of the space V_{j-1}, in particular those corresponding to the first $N_{j-1}/2$ eigenvalues,

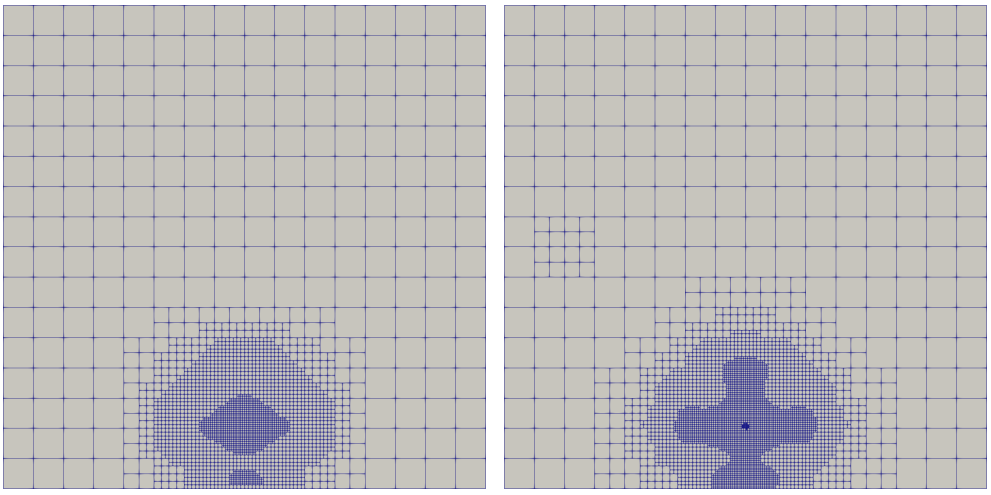


FIG. 3.1. *An example of grid refinement that fails to satisfy Assumption 3.8, by introducing a few elements with size comparable to the larger elements in the original mesh. Due to the large number of smaller elements in the original grid, these elements may generate low frequency eigenfunctions (with respect to the new grid) that cannot be captured using only the lower part of the spectrum of the original mesh.*

can be represented *exactly* by low frequencies of V_j, i.e., they should be representable as linear combinations of the first $N_j/2$ eigenfunctions of V_j.

This is false in general, and only holds up to approximation order (in an a priori setting) or up to the a posteriori error estimate for the low frequency eigenfunctions in an adaptive setting. However, it is safe to state that local refinement in finite element simulations introduces more frequencies in the higher part of the spectrum, perturbing only slightly the lowest part of the spectrum.

Theorem 3.9 should be modified to take into account that Assumption 3.8 is satisfied only approximately. The main statement would still remain valid, but we would also have higher order error terms appearing in the error propagation formula (3.25), due to the inexactness of the low frequency prolongations.

Figure 3.1 provides an example of grid refinement that may be troublesome for the above hypothesis: when passing from the left grid to the right one, we are introducing some low-frequency terms (in the middle left side of the mesh), that may invalidate the assumption. Notice that, from the practical point of view, the assumption is verified with very good accuracy whenever the size of the newly added elements remains smaller than the bottom 50% of the elements, i.e., for most refinements that do not add too many degrees of freedom between refinement levels.

Even when this fails to occur, i.e., when Assumption 3.8 is clearly not satisfied, the high frequencies that are added by the refinement are still damped very quickly by the smoothing steps nonetheless, thanks to the presence of the matrix ${M_{k+1}}^{\ell}$ in front of the propagation formula (3.25).

4. A posteriori Error Analysis. In this section we first provide an insight into classical a posteriori error estimation theory. Our focus is on residual-based a posteriori error estimators, which were historically defined and derived in terms of the discrete approximation u_h. We introduce them in Subsection 4.1. The need for accounting for inexact algebraic approximations is discussed in Subsection 4.2, where

we describe the main issues that a posteriori error analysis accounting for the algebraic error has to deal with. Our contribution in this analysis is in proving a bound on the estimator for a generic function in terms of the estimator for u_h and the corresponding algebraic error in Subsection 4.3.

4.1. Classical a Posteriori Error Estimates and Analysis. Classical a posteriori error estimation theory has been focused on measuring a suitable norm of the discretization error e_h by providing upper and lower bounds in terms of a posteriori error estimators.

By definition, "regarded as an approximation to an (unknown) suitable norm of the discretization error $\|e_h\|$, a (computable) quantity $\eta(u_h)$ is called a *posteriori error estimator* if it is a function of the known domain Ω, its boundary Γ, the right-hand side f as well as of the discrete solution u_h, or the underlying triangulation" [14].

There are two main requirements that an a posteriori error estimator $\eta(u_h)$ should satisfy, a part from being easy and cheap to compute: it has to be *reliable* in the sense of an upper bound

$$\|e_h\| \leq C_{\text{rel}}\eta(u_h) + h.o.t._{\text{rel}} \tag{4.1}$$

and *efficient* in the sense of a lower bound

$$\eta(u_h) \leq C_{\text{eff}}\|e_h\| + h.o.t._{\text{eff}}. \tag{4.2}$$

The multiplicative constants C_{rel} and C_{eff} are independent on the mesh size and h.o.t. denote oscillations of the right-hand side f, which in generic cases are of magnitudes smaller than $\|e_h\|$.

In adaptive mesh-refining procedures, a posteriori error estimators are used in the module $\mathcal{E}stimate$ of AFEM. In particular, reliability (4.1) assures enough refinement, while efficiency (4.2) prevents too much refinement.

Standard residual-based a posteriori error estimators are the most widely used for adaptive techniques. They were first introduced in the context of FEM by Babuška and Rheinboldt in [8] and they have been thereafter widely studied in the literature; we refer, e.g., to the books by Verfürth [38] and by Ainsworth and Oden [2].

Their derivations is based on the residual functional associated to the Galerkin solution, which is defined as $\mathcal{R}\{u_h\} : H_0^1(\Omega) \longrightarrow \mathbb{R}$, $\mathcal{R}\{u_h\} := (f, \bullet) - a(u_h, \bullet)$ with corresponding dual norm

$$\|\mathcal{R}\{u_h\}\|_\star := \sup_{v\in H_0^1(\Omega)\setminus\{0\}} \frac{\mathcal{R}\{u_h\}(v)}{|v|_1} = \sup_{v\in H_0^1(\Omega)\setminus\{0\}} \frac{(f,v) - a(u_h,v)}{|v|_1}. \tag{4.3}$$

The identity $|e_h|_1 = \|\mathcal{R}\{u_h\}\|_\star$ leads to reliable and efficient residual-based a posteriori bounds for the discretization error via estimation of the residual function. The main tool exploited in the derivation is the Galerkin orthogonality (2.4), which fails to be satisfied when algebraic errors are present.

4.2. A posteriori analysis with algebraic Error. When solving real-world practical applications, the main difficulty one has to face is that exact (or even near-to-exact) solutions of the algebraic problem associated to finite element problems cannot be computed. The approximation u_h^c that one obtains in a computer, does not satisfy the Galerkin property (2.4). The total error can be written as the sum of two contributions

$$\underbrace{u - u_h^c}_{\text{total error}} = \underbrace{(u - u_h)}_{\text{discretization error}} + \underbrace{(u_h - u_h^c)}_{\text{algebraic error}}. \tag{4.4}$$

The algebraic error may have a significant effect on the computed approximation, and the solution of the algebraic problem has to be considered an indivisible part of the overall solution process [34].

This issue is reflected in adaptive mesh-refining procedures. The common practice in computational sciences and engineering community has been to replace u_h by u_h^c in the expression of the error estimator η during the module $\mathcal{E}stimate$. A vast literature proposes the use of standard residual-based a posteriori error estimator on the discretization error as a basic building block and extends it, using various heuristics arguments, to incorporate the algebraic error. We refer to the seminal and investigative paper by Papež and Strakoš [33] and the references therein for various approaches.

Residual-based a posteriori error estimates for the total error for the model problem have been published in [11], [4] and [33].

In [33], the authors give the detailed proof of the residual-based upper bound on the energy norm of the total error

$$|u - v_h|_1^2 \leq 2C^2(J^2(v_h) + osc^2) + 2C_{\text{intp}}^2|u_h - v_h|_1^2, \tag{4.5}$$

for $v_h \in V_h$, with the positive multiplicative factors C and C_{intp} that are independent of u, u_h and h, but depend on the shape regularity of the triangulation. The term accounting for the algebraic error is scaled by a multiplicative factor C_{intp} that was introduced in [17]. It represents however a worst case scenario that can lead to an overestimation and it is in general not easy to estimate.

The above discussed issues make the application of the residual-based error estimator for the mesh refinement adaptivity in presence of algebraic errors an open problem, as claimed in [33]. Moreover, when considering h-adaptive algorithms, another difficulty is added: in the bound (4.5) the algebraic error is estimated globally and its local contributions cannot guarantee an indication of the spatial distribution of the discretization error over the domain (cf. [28] and [31]). In this regard, there have been recently developed flux reconstruction methodologies that introduce robust stopping criteria and balance the algebraic and discretization error; we refer to the work [34] and to the references therein for a more elaborated insight on the topic.

4.3. An upper bound on the Error Estimator applied to generic functions. We recall standard upper bounds on the discretization error and lower bounds on the total error (see [15] and [11]), and we prove an upper bound on the estimator defined for a generic finite element function $v_h \in V_h$, in terms of the estimator defined for the Galerkin solution and the algebraic error. We briefly introduce the notation that we will adopt for the estimates. Let h_T=diam(T) for $T \in \mathcal{T}_h$, h_z=diam(ω_z) for $z \in \mathcal{N}_{h,int}$, and h_E=diam(E) for $E \in \mathcal{E}_h$. Consider the mean value operator $\pi_{\omega_z} : L^1(\Omega) \longrightarrow \mathbb{R}$, $\pi_{\omega_z}(f) := \int_{\omega_z} f/|\omega_z|$.

For a given $z \in \mathcal{N}_h$, define an oscillation term

$$\text{osc}_z := |\omega_z|^{1/2}\|f - \pi_{\omega_z} f\|_{\omega_z} \quad \text{osc} := \left(\sum_{z\in\mathcal{N}_h} \text{osc}_z^2\right)^{1/2}. \tag{4.6}$$

For a given function $v_h \in V_h$, define for $E \in \mathcal{E}_h$ and $T \in \mathcal{T}_h$

$$
\begin{aligned}
J_E(v_h) &:= h_E^{1/2} \left\| \left[\frac{\partial v_h}{\partial n_E} \right] \right\|_E, \; J_T(v_h) := \sum_{E \in \partial T} J_E(v_h), \\
J(v_h) &:= \left(\sum_{E \in \mathcal{E}_h} J_E(v_h)^2 \right)^{1/2} = \left(\frac{1}{2} \sum_{T \in \mathcal{T}_h} J_T(v_h)^2 \right)^{1/2},
\end{aligned} \tag{4.7}
$$

where $[\bullet]$ is the standard notation for denoting the jump of a piecewise continuous function across the edge/face E in normal direction n_E and where we have taken into consideration that when summing overall the elements each edge/face is counted twice.

Lemma 4.1 recalls the classical upper bound on the discretization error, which is stated and proved in [15].

LEMMA 4.1 (Upper bound on the discretization error). *There exists a constant $C^\star > 0$ which depends on the shape of the triangulation, on Ω, on Γ, and which is independent of f and of the mesh-sizes h_T such that*

$$
|u - u_h|_1 \leq C^\star (osc^2 + J^2(u_h))^{1/2}. \tag{4.8}
$$

The a posteriori residual-based estimator in the rhs of (4.8) is made up by an oscillating-contribution (volume-contribution) that measures the variations of the rhs function f from its mean value $\pi_{\omega_z}(f)$ on each patch ω_z, and by an edge/face-contribution that measures the jump of the gradient of the Galerkin solution across the inner edges/faces. Notice that the global upper estimate (4.8) is made up by local cell-wise estimations.

REMARK 4.1.1. *The proof of (4.8) is based on a quasi-interpolation operator that was first introduced in [15]. It represents a modification of the classical quasi-interpolation operator due to Clément [19] in the setting of a partition of the unity, which has the effect that the volume contribution term (4.6) in the a posteriori residual based estimate (4.8) is smaller compared to the one in the standard estimate [38], [2]. The edge/face-contribution (4.7) dominates the residual based standard a posteriori estimates for affine finite element approximations [15], [17], and if the right-hand-side f is smooth, a Poincaré inequality shows that the oscillating term (4.6) is of higher order [15].*

In [11], the authors use standard bubble-function techniques of [37] to prove a global lower bound on the $|\bullet|_1$-norm distance between the true solution $u \in H_0^1(\Omega)$ and a generic function $v_h \in V_h$.

LEMMA 4.2 (Lower bound on the total error). *There exists a constant $C_\star > 0$ which only depends on the minimum angle of the triangulation, on Ω, on Γ, and which is independent of f, u, u_h and of the mesh-sizes h_T such that*

$$
J^2(v_h) \leq C_\star (|u - v_h|_1^2 + osc^2) \; \forall v_h \in V_h. \tag{4.9}
$$

Now we can use Lemma 4.1 and 4.1 to prove our main result for this section.

THEOREM 4.3. *There exist positive constants C_1, C_2, C_3 that only depend on the minimum angle of the triangulation, on Ω, on Γ, and which are independent of f, u, u_h and of the mesh-sizes h_T such that*

$$
J^2(v_h) \leq C_1 J^2(u_h) + C_2 |u_h - v_h|_1^2 + C_3 osc^2 \; \forall v_h \in V_h. \tag{4.10}
$$

Proof. For a given function $v_h \in V_h$, we decompose $u - v_h = (u - u_h) + (u_h - v_h)$ and we apply the equality $|u - v_h|_1^2 = |u - u_h|_1^2 + |u_h - v_h|_1^2$ (see, e.g. [28]) to the lower bound (4.9)

$$\begin{aligned} J^2(v_h) &\leq C_\star(|u - v_h|_1^2 + osc^2) \\ &= C_\star(|u - u_h|_1^2 + |u_h - v_h|_1^2 + osc^2) \\ &\leq C_\star(C^{\star 2}(osc^2 + J^2(u_h)) + |u_h - v_h|_1^2 + osc^2) \\ &= C_\star C^{\star 2} J^2(u_h) + C_\star |u_h - v_h|_1^2 + C_\star(C^{\star 2} + 1)osc^2 \\ &= C_1 J^2(u_h) + C_2 |u_h - v_h|_1^2 + C_3 osc^2, \end{aligned} \tag{4.11}$$

where we have used the upper bound (4.8) in (4.11). ∎

Theorem 4.3 gives an upper bound on $J^2(v_h)$ where v_h is a generic function (for instance, accounting for inexact approximations) in terms of $J^2(u_h)$, the square energy norm of the algebraic error, which is equal to $|u_h - v_h|_1^2$ and oscillation terms which only depend on the triagulation and the data, but are independent of u_h and v_h.

A related result is found in the paper [4], where the authors set the stopping criterion for the CG method by using a residual-based error estimator in the context of elliptic self-adjoint problems. They provide an upper bound on $\eta(v_h)$ in terms of $\eta(w_h)$ and $|v_h - w_h|_1$, where v_h and w_h are generic functions in V_h. However, their proof proceeds differently, and it is based on the use of the full a -posteriori error estimator, while here we prove that a similar result holds also for the case where only $J(v_h)$ is used, i.e., when only face terms are considered in the estimator.

This result, together with Theorem 3.9, gives us a sound theoretical basis for a smoothed AFEM algorithm, where the algebraic error $|u_h - v_h|_1^2$ in the intermediate steps is given explicitly by the error propagation formula (3.25).

5. S-AFEM. In this section, we introduce the smoothed AFEM algorithm (S-AFEM). To fix the ideas, we provide a small discussion with some empirical numerical evidence that justifies the use of S-AFEM in Subsection 5.1, and describe the algorithm in Subsection 5.2.

We observe that the presence of ${M_{k+1}}^\ell$ in front of the error expression in the error propagation formula (3.25) guarantees that high frequencies would be damped very quickly by the use of Richardson smoothing. On the other hand, the largest part of the low frequency error is given by the term

$$I_2^{k+1} L_2 I_1^2 \mathbf{e}_1, \tag{5.1}$$

and by the accumulation of the error in low frequency that is due to the difference between the exat algebraic solutions in the different levels

$$\sum_{j=2}^{k} I_j^{k+1} L_j \mathbf{a}_j. \tag{5.2}$$

Of all terms, the terms (5.1) that contain $\mathbf{e}_1$ could be controlled easily (and in a computationally inexpensive way), by ensuring that first iteration of AFEM is solved accurately, i.e., considering $\mathbf{e}_1 = 0$.

The remaining low frequency terms (5.2) will have in general a smaller influence on the estimator. In particular, it is still acceptable to have a large difference between u_h and u_h^ℓ (implying a large a posteriori error estimate on the algebraic approximation

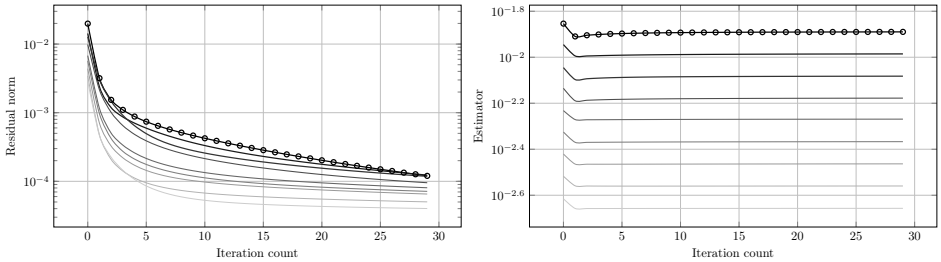


FIG. 5.1. *Algebraic residual ℓ^2-norm (left) and error estimator (right) for intermediate cycles of the classical AFEM algorithm when using Richardson iteration without preconditioner as a solver, with prolongation from the previous solution as starting guess. Darker lines correspond to earlier cycles. Only the first 30 iterations are shown.*

$J(u_h^\ell)$), provided that this difference is *roughly* equally distributed over the grid, since a (almost) constant difference between $\eta_T(u_h)$ and $\eta_T(u_h^\ell)$ for all T would result in (almost) the same cells marked for refinement.

5.1. Some Numerical Evidences. To fix the ideas, we consider the Peak Problem in two dimensions as described in Subsection 6.1, and we apply ten cycles of the classical AFEM algorithm using non-preconditioned Richardson iterations for the algebraic resolution of the system with initial guess given by the prolongation of the previous approximation for each cycle. We use standard residual-based a posteriori error estimators (4.7) which are locally defined through the jump of the gradient of the discrete approximation across the edges/faces E of the cells (cf. Section 4). In Figure 5.1 we plot the ℓ^2-norm of the residual $\mathbf{r}_k^{(\ell)} := A_k \mathbf{e}_k^{(\ell)}$ and the value of the estimator $\eta(u_k^\ell)$ for all cycles as the Richardson iteration count ℓ increases from 1 to 30.

The same behaviour is present in every refinement cycle. The residual norm shows two different speeds of convergence. The first few iterations induce a rapid drop in the residual norm (due to convergence of the highly oscillatory terms in the solution), while the second part of the iterations converge very slowly, corresponding to the convergence speed of the low frequency in the solution. The estimator, on the other hand, stagnates after very few Richardson iterations (around two or three). In other words, $J(u_h^\ell)$ is almost the same as $J(u_h)$ for $\ell \geq 3$, empirically suggesting that the error estimator (4.7) is mainly affected by the highly oscillatory components of the discrete algebraic solution u_h^ℓ, and that the estimate provided by Theorem 4.3 may be improved by exploiting the structure of the algebraic iterative solution in Richardson iteration provided by Theorem 3.9.

Although the value we plot in Figure 5.1 for the estimator is a global one, and gives no information on the distribution of the local estimator on the grid, it is a good hint that the overall behaviour of such distribution will not be changing too much after the first few Richardson iterations. We show some numerical evidence that this is actually the case in the numerical validation provided in Section 6.

Motivated by these numerical evidences and by the earlier observations, we argue that in the intermediate AFEM cycles it is not necessary to solve exactly the discrete system. What matters instead is to capture accurately the highly oscillatory components of the discrete approximation. Low frequency components *may* have an influence on the error estimator, however, this is mostly a *global* influence, that has

a small effect on the cells that will actually be marked for refinement in the $\mathcal{M}ark$ step.

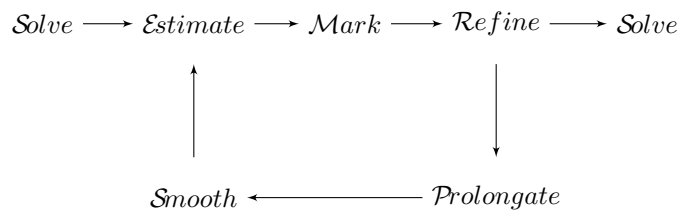
5.2. S-AFEM algorithm. We start by recalling the standard adaptive mesh-refining algorithm algorithm following [16].

Starting from an initial coarse mesh $\mathcal{T}_1$, do steps $1.-4.$ for $k = 1, \dots, \bar{k}-1$ or until criterion met

1. $\mathcal{S}olve$: $A_k \mathbf{u}_k = \mathbf{f}_k$ in $\mathbb{R}^{N_k}$, where $dim(V_k) = N_k$, based on $\mathcal{T}_k$.
2. $\mathcal{E}stimate$: Compute $\eta_T(u_k)$ for all $T \in \mathcal{T}_k$.
3. $\mathcal{M}ark$: Choose set of cells to refine $\mathcal{M}_k \subset \mathcal{T}_k$ based on $\eta_T(u_k)$.
4. $\mathcal{R}efine$: Generate a new mesh $\mathcal{T}_{k+1}$ by refinement of the cells in $\mathcal{M}_k$.

Step $k = \bar{k}$: Solve the discrete system $A_{\bar{k}} \mathbf{u}_{\bar{k}} = \mathbf{f}_{\bar{k}}$ based on $\mathcal{T}_{\bar{k}}$.
Output: nested sequence of meshes $\mathcal{T}_k$, approximations $\mathbf{u}_k$, local estimators $\eta_T(u_k)$, for $k = 1, \dots, \bar{k}-1$, and final problem-adapted approximation $\mathbf{u}_{\bar{k}}$ such that $\|\mathbf{e}_{\bar{k}}\| \leq tol$.

We present the *Smoothed Adaptive Finite Element* algorithm (S-AFEM), where the exact algebraic solution in intermediate steps is replaced by the application of a prolongation step ($\mathcal{P}rolongate$), followed by a smoothing step ($\mathcal{S}mooth$), according to the loop:



Starting from an initial coarse mesh $\mathcal{T}_1$, Solve $A_1 \mathbf{u}_1 = \mathbf{f}_1$ in $\mathbb{R}^{N_1}$ to high accuracy and generate $\mathbf{u}_1^C$. Then, do steps $1.-4.$ for $k = 2, \dots, \bar{k}-1$ or until criterion met

1. $\mathcal{S}mooth$: Compute ℓ smoothing iterations on the discrete system $A_k \mathbf{u}_k = \mathbf{f}_k$, with initial guess $\mathbf{u}_k^{(0)} := I_{k-1}^k \mathbf{u}_{k-1}^{(\ell)}$, which produce $\mathbf{u}_k^{(\ell)} \in \mathbb{R}^{N_k}$ (take $\mathbf{u}_1^{(\ell)} = \mathbf{u}_1^C$).
2. $\mathcal{E}stimate$: Compute $\eta_T(u_k^\ell)$ for all $T \in \mathcal{T}_k$.
3. $\mathcal{M}ark$: Choose set of cells to refine $\mathcal{M}_k \subset \mathcal{T}_k$ based on $\eta_T(u_k^\ell)$.
4. $\mathcal{R}efine$: Generate new mesh $\mathcal{T}_{k+1}$ by refinement of the cells in $\mathcal{M}_k$.

Step $k = \bar{k}$: Solve the discrete system $A_{\bar{k}} \mathbf{u}_{\bar{k}} = \mathbf{f}_{\bar{k}}$ to high accuracy.
Output: sequence of meshes $\mathcal{T}_k$, smoothed approximations $\mathbf{u}_k^\ell$, estimators $\eta(u_k^\ell)$, and final adapted-approximation $\mathbf{u}_{\bar{k}}^\ell$ such that $\|\mathbf{e}_{\bar{k}}\| \leq tol$.

In step $k = 1$, we capture the smoothest (i.e. less oscillatory) part of the discrete approximation by solving the discrete system exactly on the coarsest level. As the mesh is locally refined from one level to the other, we increase the higher portion of the spectrum of the matrix A_k. Thanks to the structure of the refinement in typical finite element methods, mostly high frequencies are added to the system, while low frequencies are substantially left unaltered. We formalize this by Hypothesis (3.24) on the prolongation operator for smoothed-multilevel methods.

The advantage of S-AFEM is that, on one hand, we save a substantial amount of computational time that would be needed for the algebraic solution in the intermediate steps, and on the other hand we obtain roughly the same mesh-sequence, hence the same refinement at each step, with an accuracy on the final approximation step that is comparable to the classical AFEM algorithm, at a fraction of the computational cost.

6. Numerical validation. The numerical results presented in this paper were realized using a custom `C++` code based on the `deal.II` library [9, 3, 6, 7], and on the `deal2lkit` library [36]. We consider two classical experiments used to benchmark adaptive finite element methods. A classical marking strategy is used in our implementation (see, e.g., [22]): for any level k we mark for refinement the subset of elements

$$\mathcal{M}_k := \{T \in \mathcal{T}_k : \eta_T \geq L\}, \tag{6.1}$$

where L is a *treshold error*, defined as the largest value such that

$$\Theta \sum_{T \in \mathcal{T}_k} \eta_T^2 \leq \sum_{T \in \mathcal{M}_k} \eta_T^2. \tag{6.2}$$

The parameter Θ is such that $0 \leq \Theta \leq 1$, where $\Theta = 1$ corresponds to an almost uniform refinement, while $\Theta = 0$ corresponds to no refinement. In our numerical tests, unless otherwise stated, we set $\Theta = 0.3$. The refinement strategy that we adopt in this work is based on the use of "hanging nodes" (see [9] for a detailed discussion on the implementation details).

6.1. Two-dimensional examples.

Smooth domain, peak right hand side. The first example we consider consists in solving the model problem on a square domain, with a custom forcing term that contains a peak in a specified point in the domain, forcing the exact solution to be

$$u(x, y) = x(x-1)y(y-1)e^{-100\left((x-0.5)^2+(y-0.117)^2\right)}, \tag{6.3}$$

as shown in Figure 6.1.

L-shaped domain, smooth right hand side. In the second two-dimensional test case, we consider a L-shaped domain, i.e., a square where the upper right corner is removed, and the reentrant corner coincides with the origin. No forcing term is added to the problem, but the boundary conditions are set so that the following exact solution is obtained (when expressed in polar coordinates)

$$u(r, \theta) = r^{2/3} \sin\left(\frac{2\theta + 5\pi}{3}\right), \tag{6.4}$$

as shown in Figure 6.2.

In both cases, we apply ten cycles of classical AFEM and of S-AFEM, respectively. For the AFEM algorithm, we use the CG method as iterative solver, with an algebraic multigrid preconditioner (AMG), and we iterate until the ℓ^2-norm of the residual is below a tolerance of 10^{-12} for each cycle. For S-AFEM, we modify the intermediate cycles and we only apply only three Richardson iterations. For reference, we report a comparison between the cells marked for refinement by AFEM and S-AFEM after

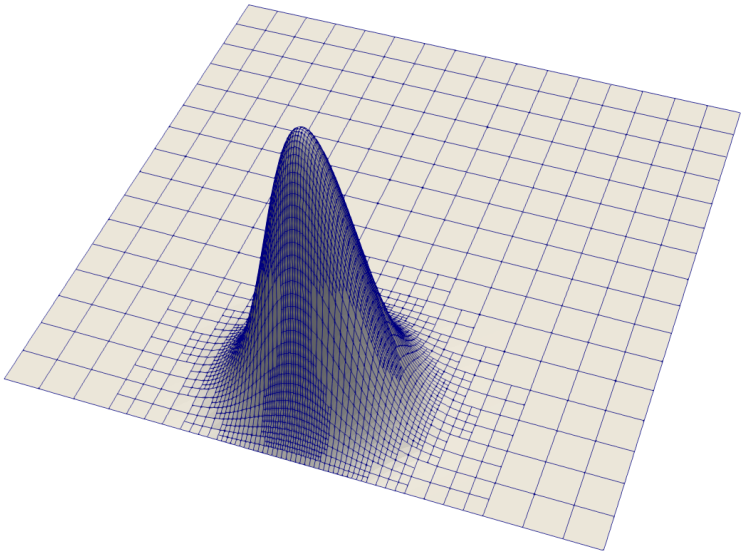


FIG. 6.1. *Solution to the Peak Problem in 2D (6.3).*

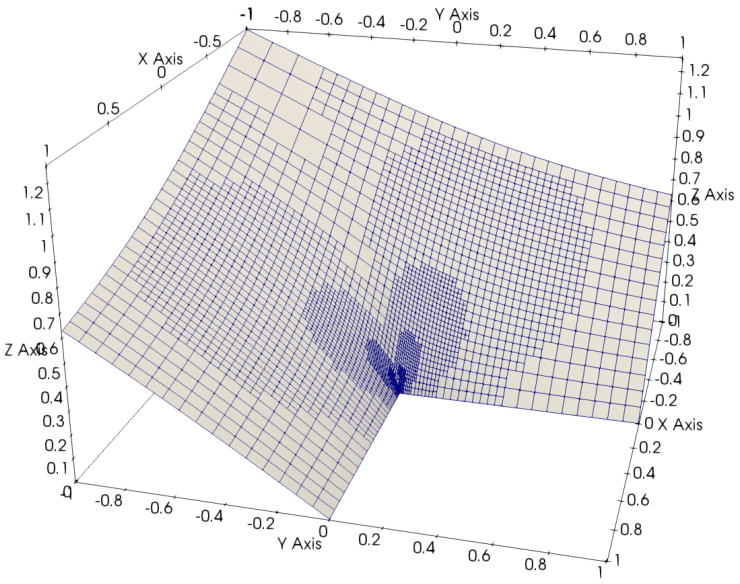


FIG. 6.2. *Solution to the L-shaped domain Problem in 2D (6.4).*

four cycles for the Peak Problem in Figure 6.3 and after nine cycles for the L-shaped domain Problem in Figure 6.4.

In both cases, the set of marked cells, although different in some areas, produces a refined grid that is very similar between the classical AFEM and the S-AFEM, and

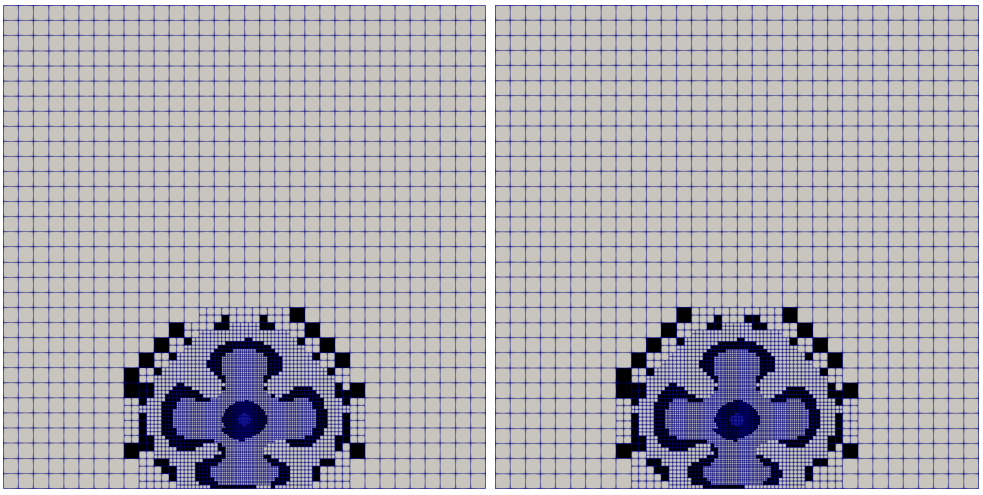


FIG. 6.3. *Comparison between the cells marked for refinement in AFEM (left) and S-AFEM (right) after* 9 *cycles for the Peak Problem in* 2*D.*

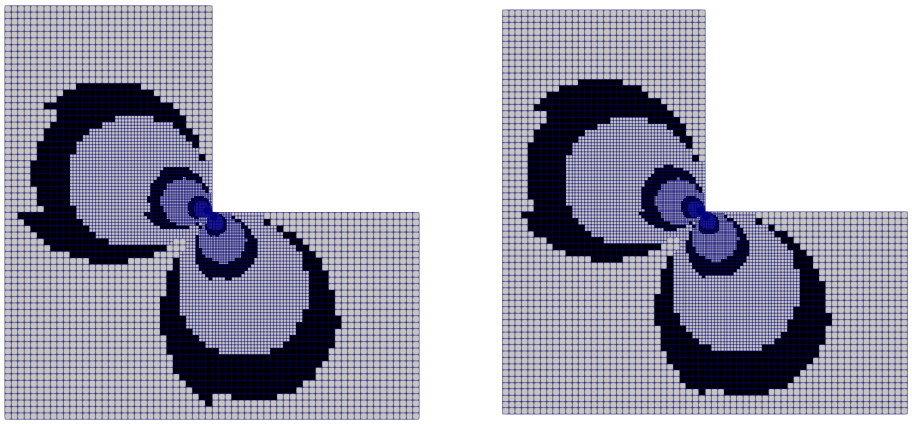


FIG. 6.4. *Comparison between the cells marked for refinement in AFEM (left) and S-AFEM (right) after* 5 *cycles for the L-shaped domain Problem in* 2*D.*

where the accuracy of the final solution is comparable.

In Figures 6.5 and 6.7 we compare the values of the global estimators $J(u_h)$ and $J(u_h^\ell)$ and of the H_1 semi-norm of the total errors for each cycle for the Peak Problem, and for the L-shaped domain Problem respectively, when using S-AFEM. For reference, Figures 6.6 and 6.8 show the error and the estimator in the classical AFEM algorithm for the two examples. Notice that the first step of AFEM and of

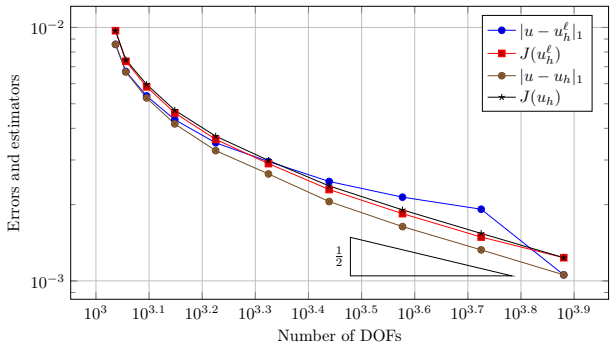


FIG. 6.5. *Values of the total error H^1 semi-norm and of the error estimator for each loop of the classical AFEM ($|u-u_h|_1$ and $J(u_h)$) and S-AFEM with $\ell = 3$ smoothing iterations ($|u-u_h^\ell|_1$ and $J(u_h^\ell)$) for the Peak Problem in 2D.*

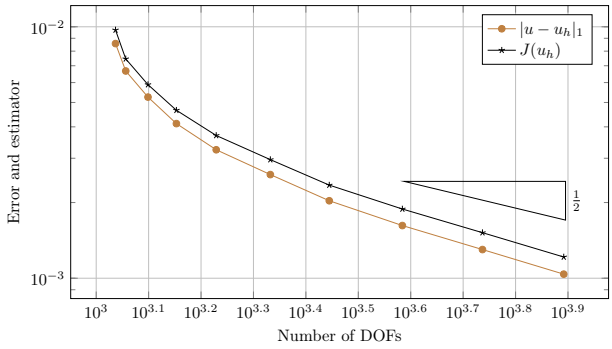


FIG. 6.6. *Values of the total error $|u-u_h|_1$ and error estimator $J(u_h)$ for the Peak Problem in 2D, using classical AFEM.*

S-AFEM are the same. The last step in the S-AFEM case shows comparable results as in the AFEM algorithm for both examples.

Notice that in S-AFEM the value of the global estimator is almost the same of the one that would be obtained by solving using CG and AMG ($J(u_h)$ in Figures 6.5 and 6.7), showing that in the two dimensional setting the error estimator (4.7) is mainly affected by the high frequencies of the discrete solution, which are well captured with just a few Richardson iterations. On the other hand, the total error increases in the intermediate cycles, due to the algebraic error that has been accumulated by applying smoothing iterations instead of solving the algebraic problem until convergence, as quantified by Theorem 3.9. This error measures the distance between the exact algebraic solution and the smooth non-oscillatory components of the approximate solution that are not captured by Richardson iteration, and have little or no influence on the error estimator, and therefore on the generated grid. After ten cycles, we solve the algebraic problem until converge using CG and AMG, as in the first cycle, and we obtain a solution whose error is controlled by the estimator, as expected.

6.2. Three-dimensional examples.

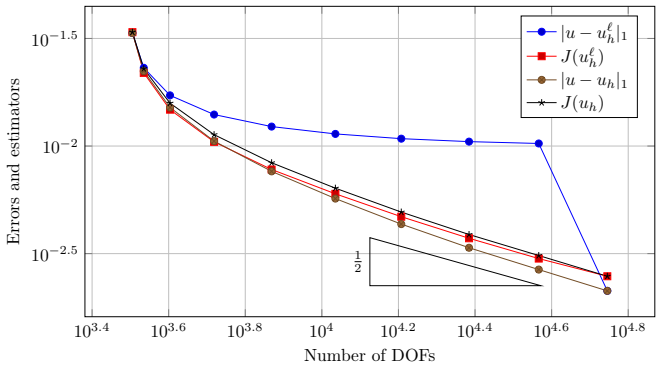


FIG. 6.7. *Values of the total error H^1 semi-norm and of the error estimator for each loop of the classical AFEM ($|u-u_h|_1$ and $J(u_h)$) and S-AFEM with $\ell = 3$ smoothing iterations ($|u-u_h^\ell|_1$ and $J(u_h^\ell)$) for the L-shaped domain Problem in 2D.*

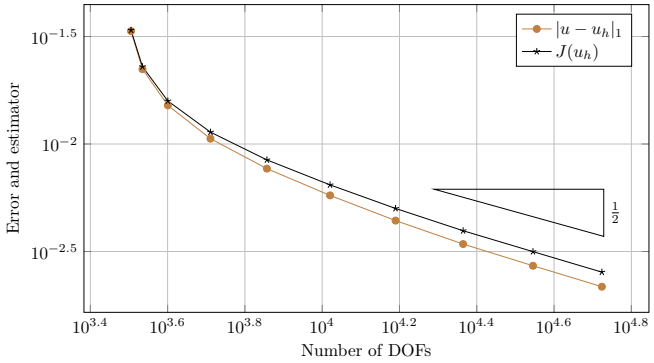


FIG. 6.8. *Values of the total error $|u-u_h|_1$ and error estimator $J(u_h)$ for the L-shaped domain Problem in 2D, using classical AFEM.*

Smooth domain, peak right hand side. The first three-dimensional test case that we propose is a model problem on a cube domain, where the forcing term contains a peak in a specified point that forces the exact solution to be given by

$$u(x,y,z) = x(x-1)y(y-1)z(z-1)e^{-100\left((x-0.5)^2+(y-0.117)^2+(z-0.331)^2\right)} \tag{6.5}$$

as shown in Figure 6.9.

Fichera corner domain, smooth right hand side. In the second three-dimensional example, we consider the classic Fichera corner domain, i.e., a cube where the upper right corner is removed, and the reentrant corner coincides with the origin. We set the exact solution to be

$$u(r,\theta,\phi) = r^{1/2}, \tag{6.6}$$

and we add a forcing term that induces the above exact solution as shown in Figure 6.10.

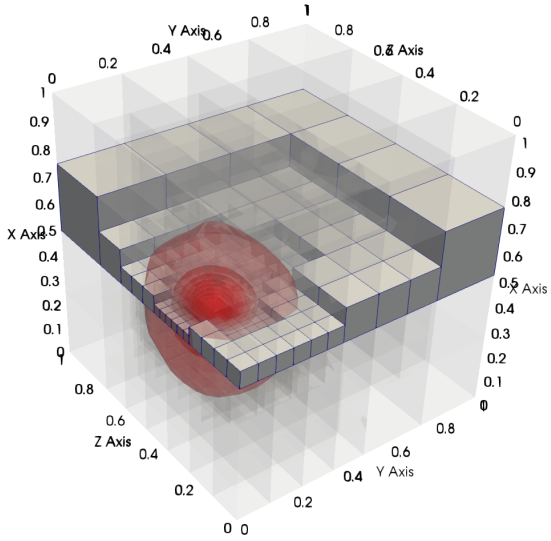


FIG. 6.9. *Solution to the Peak Problem* (6.5) *in 3D.*

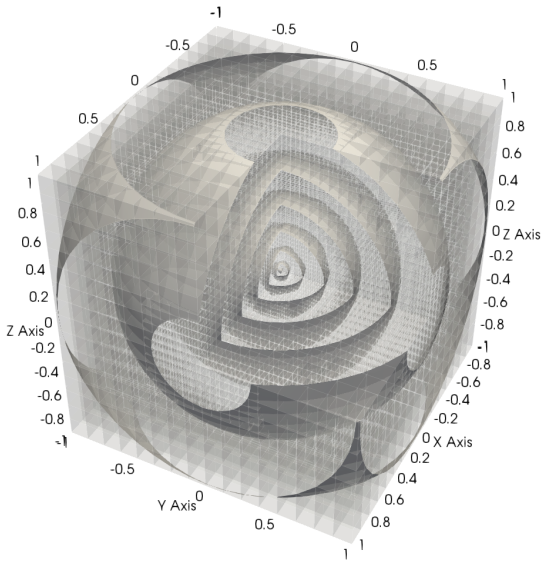


FIG. 6.10. *Solution to the Fichera domain Problem* (6.6) *in* 3*D.*

In both examples, the estimator applied to the algebraic solution after three smoothing steps (see Figures 6.11 and 6.13) seems to be more sensitive to the low frequency content of u_h^ℓ.

For reference, Figures 6.12 and 6.14 show the error and the estimator in the classical AFEM algorithm for the two examples. In other words, in the three-dimensional case the combination of Theorems 3.9 and 4.3 provides a sharper estimate. This may

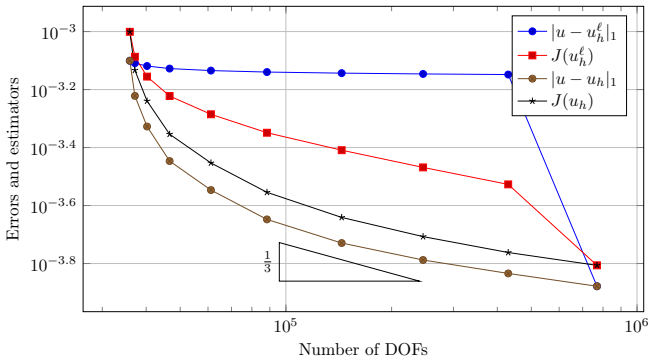


FIG. 6.11. *Values of the total error H^1 semi-norm and of the error estimator for each loop of the classical AFEM ($|u-u_h|_1$ and $J(u_h)$) and S-AFEM with $\ell = 3$ smoothing iterations ($|u-u_h^\ell|_1$ and $J(u_h^\ell)$) for the Peak Problem in 3D.*

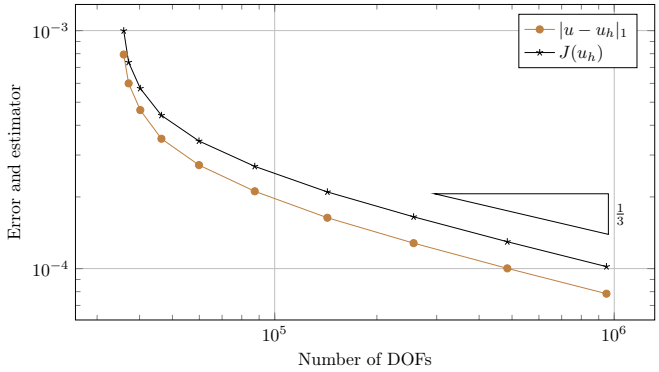


FIG. 6.12. *Values of the total error $|u-u_h|_1$ and error estimator $J(u_h)$ for the Peak Problem in 3D, using classical AFEM.*

also be related to the fact that the increase on the number of degrees of freedom between successive cycles in the three-dimensional setting is much more severe w.r.t. the two-dimensional case, maybe hindering the non-interacting frequency coupling hypothesis. Nonetheless, the difference in accuracy at the final step between AFEM and S-AFEM is negligible also in the three-dimensional case, showing that the (small) differences in the refinement patterns between AFEM and S-AFEM do not influence the final accuracy.

6.3. Computational costs. In the following table we show a comparison of the computational cost associated to the classical AFEM and to the smoothed AFEM, for the four examples we presented in the previous section.

The results were obtained on a 2.8 GHz Intel Core i7 with 4 cores and 16GB of RAM, using MPI parallelization on all 4 cores.

Table 6.1 only shows the comparison between AFEM and S-AFEM in the solve phase, where S-AFEM is always faster than AFEM, offering an average speedup of a factor three. In the table we compare the computational cost of all intermediate cycles in S-AFEM (Intermediate solves (Richardson) in the table), with the corresponding computational cost for standard AFEM (Intermediate solves (CG) in the table). The

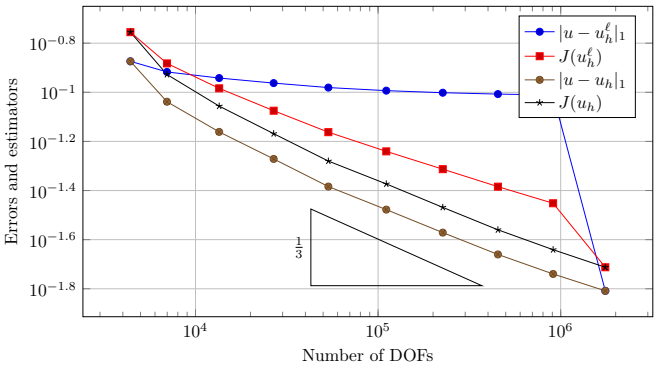


FIG. 6.13. *Values of the total error H^1 semi-norm and of the error estimator for each loop of the classical AFEM ($|u-u_h|_1$ and $J(u_h)$) and S-AFEM with $\ell = 3$ smoothing iterations ($|u-u_h^\ell|_1$ and $J(u_h^\ell)$) for the Fichera corner domain Problem in 3D.*

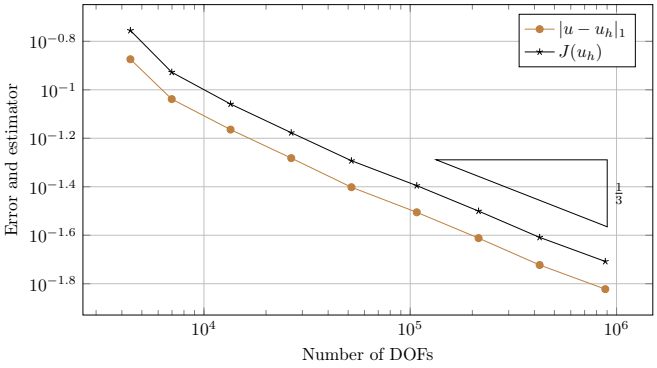


FIG. 6.14. *Values of the total error $|u-u_h|_1$ and error estimator $J(u_h)$ for the Fichera corner Problem in 3D, using classical AFEM.*

first and last solve are the same in the two algorithms, and are reported to provide a scaling with respect to the total computational cost of the solution phase in the program. Other phases (like graphical output, mesh setup, assembling setup, and error estimation) are not shown since they are identical in the two algorithms.

7. Conclusions. In this work we propose a new smoothed algorithm for adaptive finite element methods (S-AFEM), inspired by multilevel techniques. In S-AFEM, the classical algorithm of AFEM (*Solve-Estimate-Mark-Refine*) is modified to replace the *Solve* step in intermediate cycles by successive applications of *Prolongate* and *Smooth* steps, where the solution from the previous cycle is transferred to the current grid, and a fixed number of smoothing iterations are applied to obtained an inexact (but cheap to compute) approximation of the algebraic solution.

We analyzed the error propagation properties of the S-AFEM algorithm, and provided a bound on the a-posteriori error estimator applied to the approximated algebraic solution. Although the results are not sharp, they provide a good insight on why the S-AFEM algorithm is capable of producing a mesh sequence that is very close to the one obtained by classical AFEM, at a fraction of the computational cost.

An interesting question for future investigations is whether this technique may be applied to more complex second-order elliptic problems, and whether one can obtain

	Peak 2D	L-shaped 2D	Peak 3D	Fichera 3D
First and last solve	0.0187s	0.0601s	32s	101s
Intermediate solves (CG)	0.0663s	0.219s	76.4s	185s
Intermediate solves (Richardson)	0.005s	0.00892s	0.252s	0.426s
S-AFEM intermediate speedup	13.26	24.6	303.7	434.3
S-AFEM total speedup	3.59	4.045	3.361	2.819

TABLE 6.1

Comparison of the computational cost of the solution stage for ten cycles of adaptive refinement using classical AFEM and S-AFEM.

better results with more articulated smoothing algorithms.

REFERENCES

[1] R. A. Adams and J. J. Fournier. *Sobolev spaces*, volume 140. Academic press, 2003.

[2] M. Ainsworth and J. T. Oden. *A posteriori error estimation in finite element analysis*, volume 37. John Wiley & Sons, 2011.

[3] G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmoeller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells. The `deal.II` library, version 9.0. *Journal of Numerical Mathematics*, 26(4):173–183, 2018.

[4] M. Arioli, E. H. Georgoulis, and D. Loghin. Stopping criteria for adaptive finite element solvers. *SIAM Journal on Scientific Computing*, 35(3):A1537–A1559, 2013.

[5] M. Arioli, J. Liesen, A. Miçdlar, and Z. Strakoš. Interplay between discretization and algebraic computation in adaptive numerical solutionof elliptic pde problems. *GAMM-Mitteilungen*, 36(1):102–129, 2013.

[6] D. Arndt, W. Bangerth, T. C. Clevenger, D. Davydov, M. Fehling, D. Garcia-Sanchez, G. Harper, T. Heister, L. Heltai, M. Kronbichler, R. Maguire Kynch, M. Maier, J. P. Pelteret, B. Turcksin, and D. Wells. The deal.II Library, Version 9.1. *Journal of Numerical Mathematics*, 2019.

[7] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells. The deal.ii finite element library: Design, features, and insights. *Computers and Mathematics with Applications*, 2020.

[8] I. Babuška and W. C. Rheinboldt. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615, 1978.

[9] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.

[10] R. Becker, C. Johnson, and R. Rannacher. Adaptive error control for multigrid finite element. *Computing*, 55(4):271–288, 1995.

[11] R. Becker and S. Mao. Convergence and quasi-optimal complexity of a simple adaptive finite element method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43(6):1203–1219, 2009.

[12] J. H. Bramble. *Multigrid methods*. Routledge, 2018.

[13] J. H. Bramble and X. Zhang. The analysis of multigrid methods. *Handbook of numerical analysis*, 7:173–415, 2000.

[14] S. C. Brenner and C. Carstensen. Finite element methods. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–47, 2017.

[15] C. Carstensen. Quasi-interpolation and a posteriori error analysis in finite element methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33(6):1187–1202, 1999.

[16] C. Carstensen, M. Feischl, M. Page, and D. Praetorius. Axioms of adaptivity. *Computers & Mathematics with Applications*, 67(6):1195–1253, 2014.

[17] C. Carstensen and R. Verfürth. Edge residuals dominate a posteriori error estimates for low order finite element methods. *SIAM journal on numerical analysis*, 36(5):1571–1587, 1999.

[18] P. G. Ciarlet. The finite element method for elliptic problems. *Classics in applied mathematics*, 40:1–511, 2002.

[19] P. Clément. Approximation by finite element functions using local regularization. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):77–84, 1975.

[20] P. Daniel, A. Ern, and M. Vohralík. An adaptive hp-refinement strategy with inexact solvers and computable guaranteed bound on the error reduction factor. *Computer Methods in Applied Mechanics and Engineering*, 359:112607, 2020.
[21] P. Daniel and M. Vohralík. Guaranteed contraction of adaptive inexact *hp*-refinement strategies with realistic stopping criteria. 2020.
[22] W. Dörfler. A convergent adaptive algorithm for poisson's equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, 1996.
[23] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative schwarz algorithms. *Numerische Mathematik*, 70(2):163–180, 1995.
[24] W. Hackbusch. *Iterative solution of large sparse systems of equations*, volume 95. Springer, 1994.
[25] W. Hackbusch. *Multi-grid methods and applications*, volume 4. Springer Science & Business Media, 2013.
[26] P. Jiránek, Z. Strakoš, and M. Vohralík. A posteriori error estimates including algebraic error and stopping criteria for iterative solvers. *SIAM Journal on Scientific Computing*, 32(3):1567–1590, 2010.
[27] S. Lang. *Introduction to linear algebra*. Springer Science & Business Media, 2012.
[28] J. Liesen and Z. Strakos. *Krylov subspace methods: principles and analysis*. Oxford University Press, 2013.
[29] G. Mallik, M. Vohralík, and S. Yousef. Goal-oriented a posteriori error estimation for conforming and nonconforming approximations with inexact solvers. *Journal of Computational and Applied Mathematics*, 366:112367, 2020.
[30] A. Miraçi, J. Papež, and M. Vohralík. A multilevel algebraic error estimator and the corresponding iterative solver with *p*-robust behavior. 2019.
[31] J. Papež, J. Liesen, and Z. Strakoš. Distribution of the discretization and algebraic error in numerical solution of partial differential equations. *Linear Algebra and its Applications*, 449:89–114, 2014.
[32] J. Papež, U. Rüde, M. Vohralík, and B. Wohlmuth. Sharp algebraic and total a posteriori error bounds for h and p finite elements via a multilevel approach. 2017.
[33] J. Papež and Z. Strakoš. On a residual-based a posteriori error estimator for the total error. *IMA Journal of Numerical Analysis*, 2017.
[34] J. Papež, Z. Strakoš, and M. Vohralík. Estimating and localizing the algebraic and total numerical errors using flux reconstructions. *Numerische Mathematik*, 138(3):681–721, 2018.
[35] Y. Saad. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.
[36] A. Sartori, N. Giuliani, M. Bardelloni, and L. Heltai. deal2lkit: A toolkit library for high performance programming in deal.ii. *SoftwareX*, 7:318–327, 2018.
[37] R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. *Journal of Computational and Applied Mathematics*, 50(1-3):67–83, 1994.
[38] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh-refinement techniques*. John Wiley & Sons Inc, 1996.
[39] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM review*, 34(4):581–613, 1992.